

---

# BEST PRACTICES PROPOSED BY IGNITION

---

DAQ SCWG Meeting  
11-March-2021  
Patricia Mendez

---

---

# BEST PRACTICES: WHY NOW

---

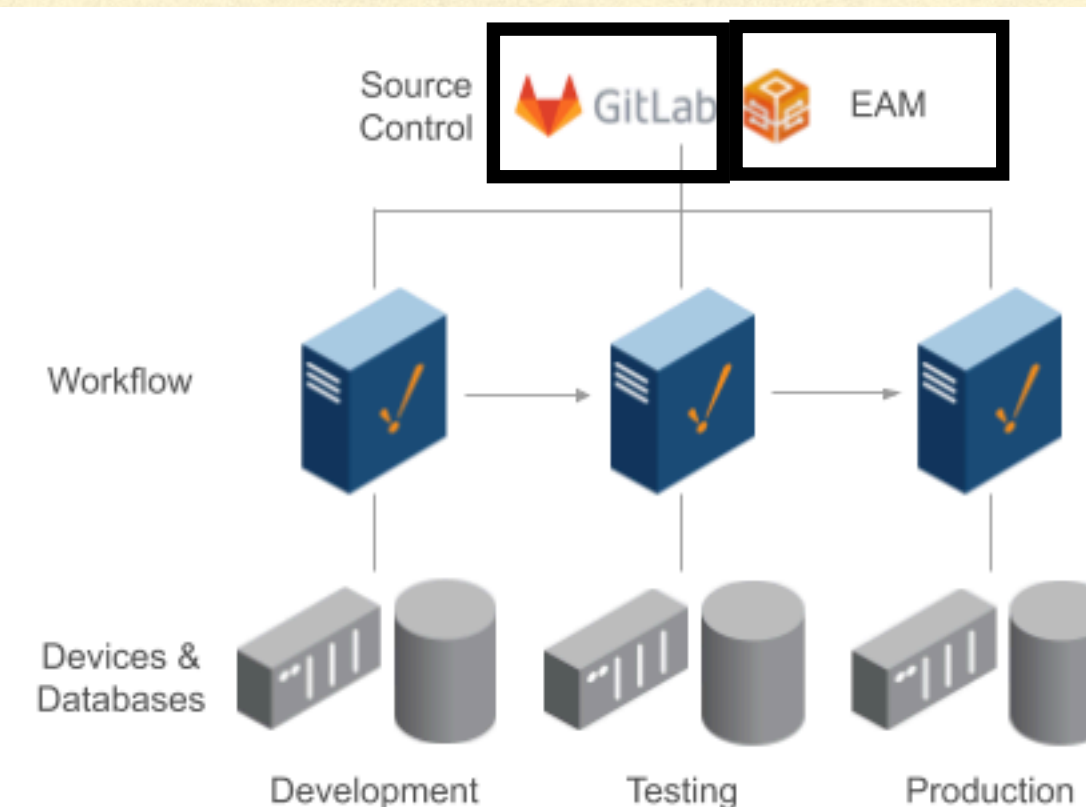
Basically because of one fundamental reason: we need to share the work we are doing across the ocean

---

# STANDARD APPROACH BY IGNITION

- **Deployment workflow:** Ignition declares a best practice set based on 3 levels of development grades

- Development
- Testing
- Production



- EAM: Enterprise Administrator Module of Ignition which helps centralising operation actions among gateways
- GitLab version control sw recommended by Ignition BUT NOT interfaced nor embedded in the tool

# IGNITION CONFIGURATION

- Ignition configuration is based on a server structure

- The server keeps the full configuration including: installation, licences, backup, connections, DB access, ACLs
- The deployment model we chose needs to focus on the server

## Gateway configuration: System core

- Settings stored inside Ignition's internal SQLite DB
- Outside the gateway backup no way to export settings among gateways: Manual approach
- Ignition proposal: Create your own documentation

- **Main areas of configuration**

## Tags

- Stored inside Tag Providers, not part of projects and they are exportable
- Local tag providers stored inside Ignition's SQLite DB
- Ignition recommends to regularly export tags in JSON format (automatic approach supported based in python scripting) to stored them in git OR use EAM to move tags among gateways

## Images

- Uploadable in the image management tool (Designer) for importing in Vision/Perspective
- Images stored inside Ignition's SQLite DB
- They cannot be automatically exported as in the case of tags
- EAM does not support the exporting of images among gateways
- Fully manual approach

## Projects: Main unit of configuration

- Projects are stored in a file system so they can be tracked outside via git
  - `/var/lib/ignition/data/projects`
- EAM support projects to import/export them among gateways

# SUMMARY OF THE AREAS OF CONFIGURATION

Change Track Summary

	Gateway Configuration	Tags	Images	Projects
Change Tracking	Manual Documentation and Gateway Backup, then commit to Version Control	Tag Export, then commit to Version Control	Image Export, then commit to Version Control	No export required. Auto-commit project folder changes to Version Control
Backups	Gateway Backup	Gateway Backup	Gateway Backup	Gateway Backup

---

# DEPLOYMENT ENVIRONMENTS: DEVELOPMENT LEVEL, OUR CURRENT SCENARIO

---

- Ignition observes 2 scenarios in this case:
    - Shared development area with a central gateway dedicated to accessed via ACLs
    - Individual workstations (this is our model) where the use of a git like structure becomes even more important
  - Ignition licence proposed in this level: Trial 2h
  - Simulation of real devices: Tricky part when it has to coexist with a production one
    - OPC UA: multiple connections are possible (watch out writing approaches via tags)
    - MQTT: leave the publication responsibility to the production environment and subscribe from any other one (again take care of writing procedures via tags)
    - Gateway Network: implement the use of remote tag provides to sharing tags information
    - PLC's simulators are supported and provided in Ignition allowing the definition of any tag path. If you keep the same name on the devices, the tags will be identical across the systems
-

---

# EAM: ENTERPRISE ADMINISTRATOR MODULE

---

- Intuitive way of managing several ignition installations across multiple gateways
  - Ignition proposes a model based on a central gateway acting as a central controller
    - Check health and diagnosis
    - New modules deployment
    - licences handling
    - remote backups
  - Good for moving tags but limited for the rest of components
-

---

# CONCLUSIONS AND OUR POSSIBLE SCENARIO

---

- **Conclusions**

- Ignition: Good to use gitlab, do it, but no interface from the tool.
- EAM: From my point of view quite limited for our use case (basically interesting for Tags only and this can be done a la WinCC OA exporting them to JSON files)

- **Our first steps towards a common effort**

- Set a GitHub common structure (DUNE-IGNITION maybe?)
  - Establish different levels of recording
    - gateway backups mostly for archiving purposes
    - projects
    - tags —> set remote tag providers simultaneously
-